# ConVAE Results

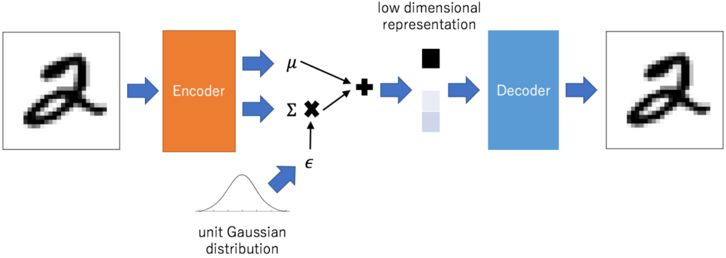
Breno Orzari

Sprace

# Contents

# Variational Autoencoder (VAE)

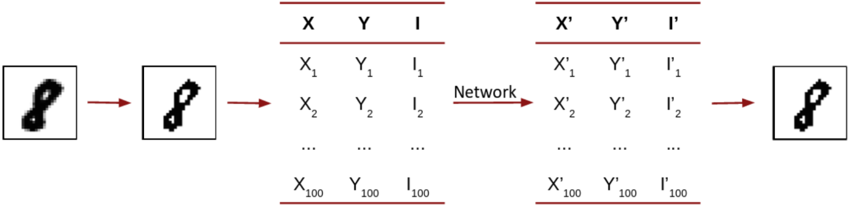
- A Variational Autoencoder is a neural network that has the following structure:




- Its main feature is the capability of generating new outputs through the encoding of the training inputs into a low dimensional representation
- The cost function is given by two terms:
  - The reconstruction loss term and a KL divergence

# Sparse MNIST Dataset, Motivation and Goal

- The sparse MNIST dataset takes only the 100 most intense pixels from the standard MNIST dataset digits:




- What is fed into the network is a matrix that has the position (x, y) and the intensity (I) of those pixels
- The motivation to use this dataset comes from the particle detectors calorimeters:
  - A particle signal in a calorimeter is given by its position $(\eta, \phi)$ and its transverse momentum $(p_T)$
- The goal is to build a ConVAE capable of generating sparse MNIST digits

# Sparse Reconstruction Loss Term

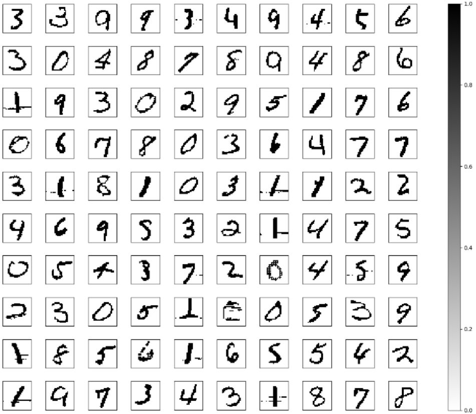
- The first step was to test a different reconstruction loss term to be applied in the sparse data:

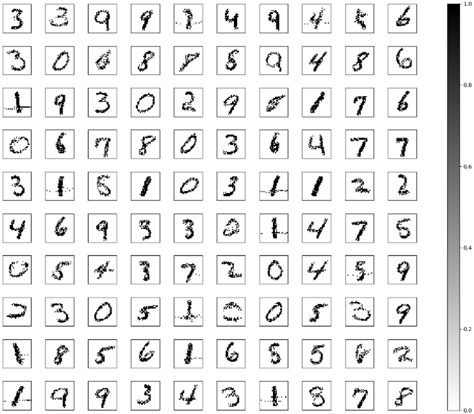
$$L_{rec} = \sum_i min\ [d_E(x, \hat{x}_i)]^2 + \sum_i min\ [d_E(x_i, \hat{x})]^2 \quad (1)$$

- It gets the distance between the closest output pixel to a given input pixel and sums with the distance between the closest input pixel to a given output pixel, of every input/output pixels of an image
- Only considering pixels positions for now (the intensities different than 0 were set to be 1.0)

# Sparse Reconstruction Loss Term

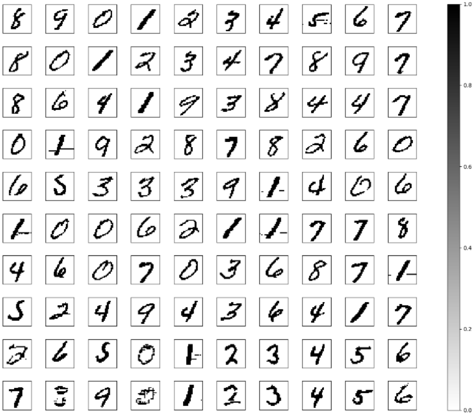
- Reconstruction of training images

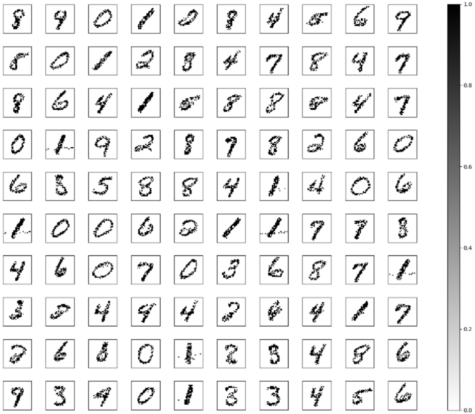


# Sparse Reconstruction Loss Term

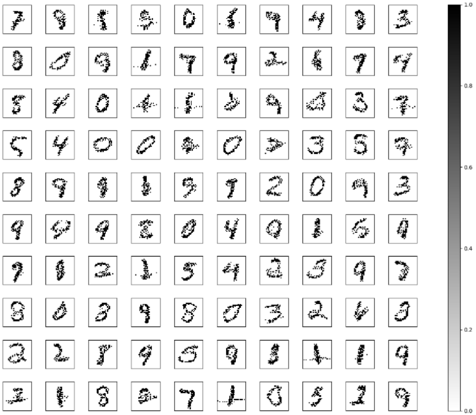
- Reconstruction of test images





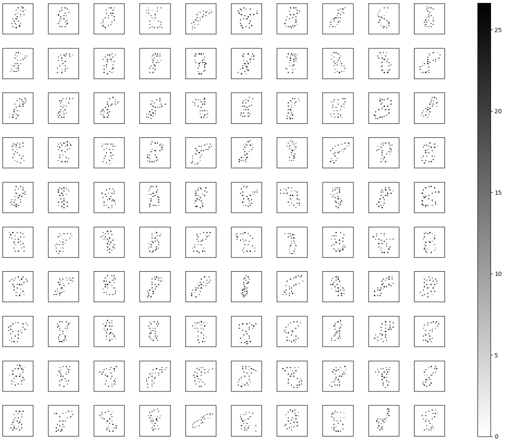
# Sparse Reconstruction Loss Term

- Generation of images



# MNIST Superpixels Dataset

- Only 25 pixels with intensities greater than 0
- Very sparse
- Using only the digit 8
  - The idea would be to train other networks for other digits



# Sparse Reconstruction Loss Term

- Test two different approaches
  - Suggested by a member of the group:

$$L_{rec} = \sum_i min\ [d_E(x_i,\ \hat{x})]^2 + \sum_i min\ [d_E(x,\ \hat{x}_i)]^2 + \sum_i (I_i - \hat{I}_{\hat{k}})^2 + \sum_i (I_k - \hat{I}_i)^2 \quad (2)$$

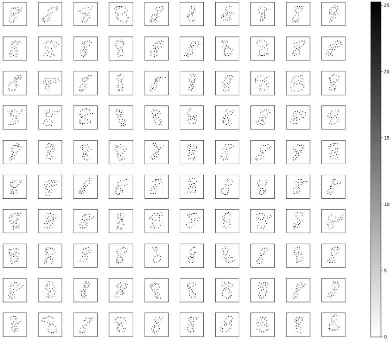
    - $x = (x,\ y)$ and $\hat{x} = (\hat{x},\ \hat{y})$ are the positions of input and output pixels respectively;
    - $d_E$ is the euclidean distance;
    - $I$ and $\hat{I}$ are the intensities of input and output pixels respectively;
    - $\hat{k} =\ argmin\ [d_E(x_i,\ \hat{x})]^2$ and $k =\ argmin\ [d_E(x,\ \hat{x}_i)]^2$;
  - Use the pixel intensity as the third axis of a 3D space:

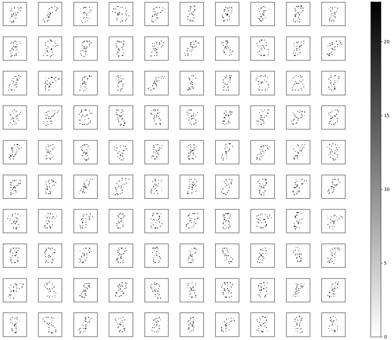
$$L_{rec} = \sum_i min\ [d_E(p_i,\ \hat{p})]^2 + \sum_i min\ [d_E(p,\ \hat{p}_i)]^2 \quad (3)$$

    - $p = (x,\ y,\ I)$ and $\hat{p} = (\hat{x},\ \hat{y},\ \hat{I})$ are input and output pixels features respectively;

# Sparse Reconstruction Loss Term

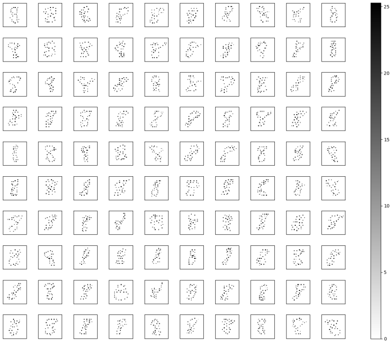
- Reconstruction of training images: first approach

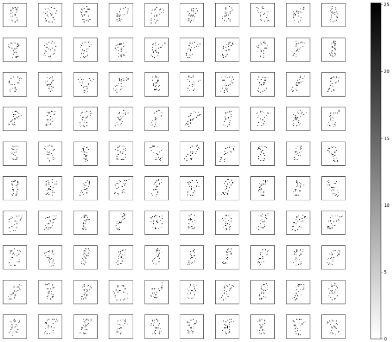


# Sparse Reconstruction Loss Term

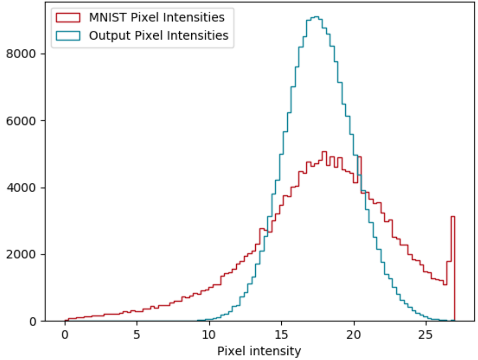
- Reconstruction of training images: second approach



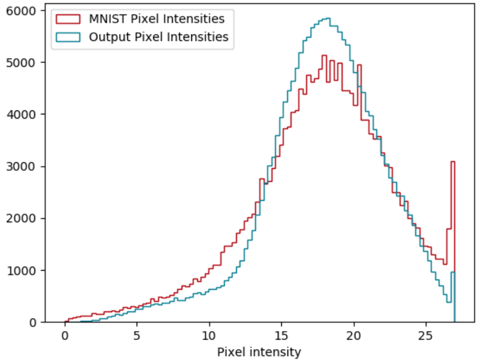
# Sparse Reconstruction Loss Term

- Pixel intensity comparison: first approach



# Sparse Reconstruction Loss Term

- Pixel intensity comparison: second approach



# Jets Dataset

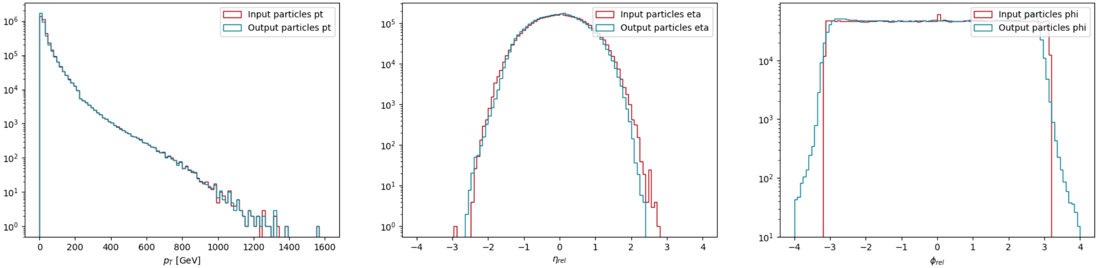
- 5 categories of jets: gluons, quarks, W bosons, Z bosons and top jets;
- Each jet is a list of particles (30p, 50p, 100p or 150p), with their respective $p_T$, $\eta$ and $\phi$, ordered by decreasing $p_T$;
- The goal is to train a ConVAE on this dataset to be able to generate jets;
- The loss function being used is:

$$L_{rec} = \sum_i min\ [d_E(p_i,\ \hat{p})]^2 + \sum_i min\ [d_E(p,\ \hat{p}_i)]^2 \tag{4}$$

  - where $p = (p_T,\ \eta,\ \phi)$ and $\hat{p} = (\hat{p_T},\ \hat{\eta},\ \hat{\phi})$ are input and output particles features respectively, and $d_E$ is the Euclidean distance;

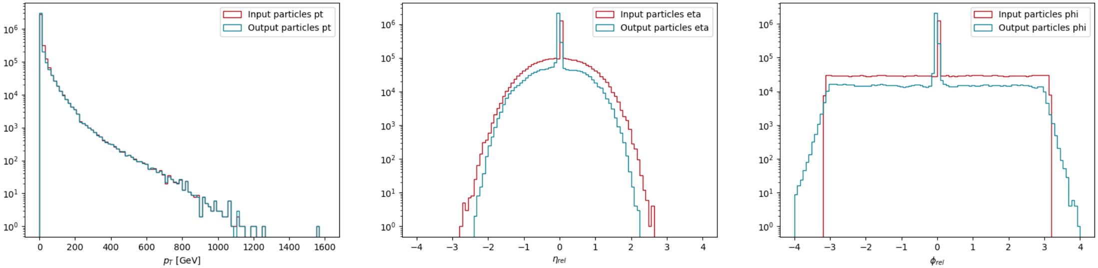
# Latest Results

- 30p gluon jets



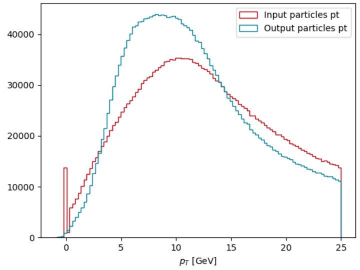
# Latest Results

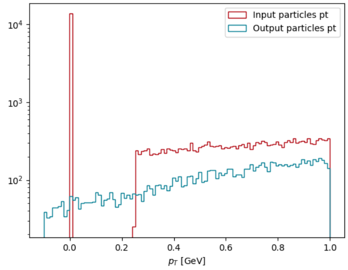
- 100p gluon jets



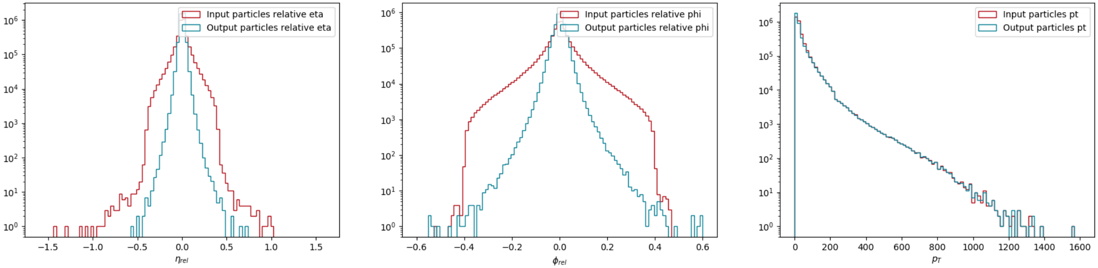
# Latest Results

- Three problems appeared:
  - ghost particles have $p_T$, $\eta$ and $\phi$ are zero $\rightarrow$ we could pad them away from the particles in the jets;
  - how to handle the periodicity of $\phi$ $\rightarrow$ use the relative quantities $\eta_{rel}$ and $\phi_{rel}$;
  - low $p_T$ reconstruction (images below);



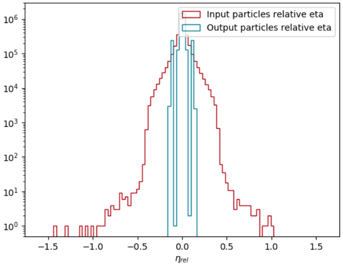



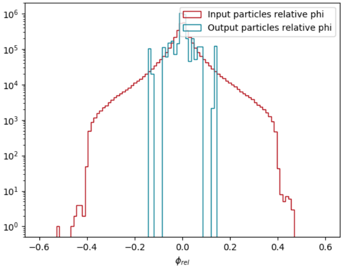
- 30p gluon jets

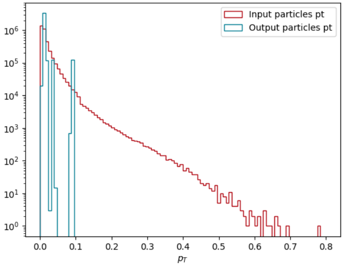


# 30p relative features: dividing $p_T$ by 2000

- 30p gluon jets

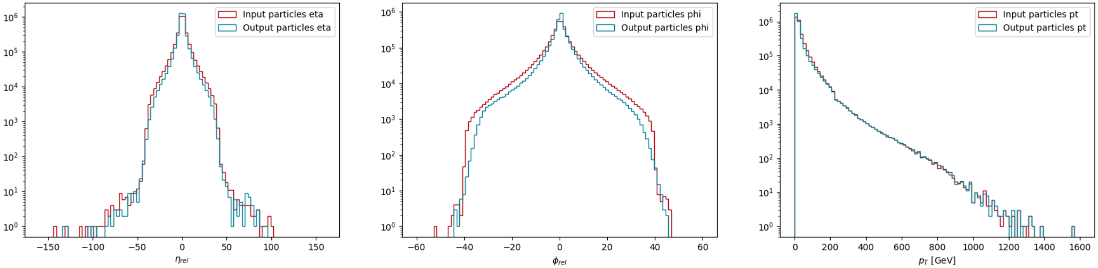


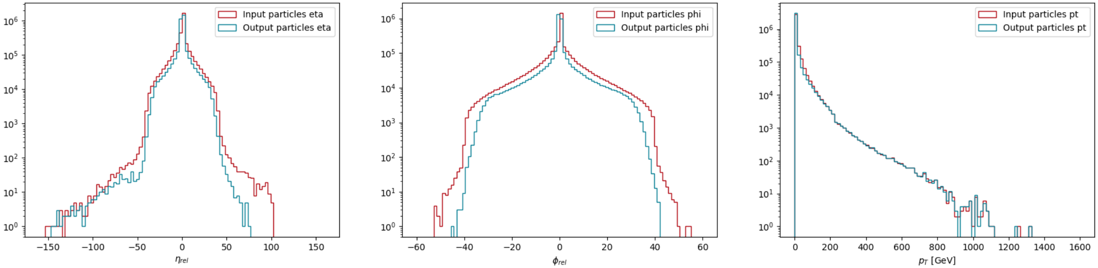
# 30p relative features: multiplying $\eta_{rel}$ and $\phi_{rel}$ by 100

- 30p gluon jets



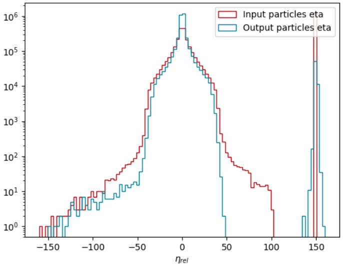
# 100p relative features: multiplying $\eta_{rel}$ and $\phi_{rel}$ by 100, without padding

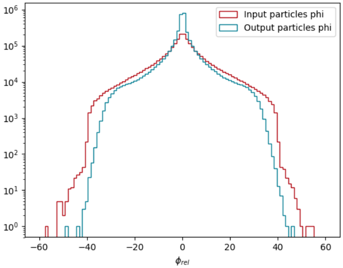
- 100p gluon jets

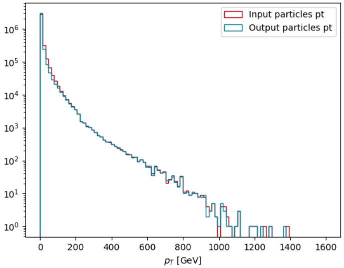


# 100p relative features: multiplying $\eta_{rel}$ and $\phi_{rel}$ by 100, with padding

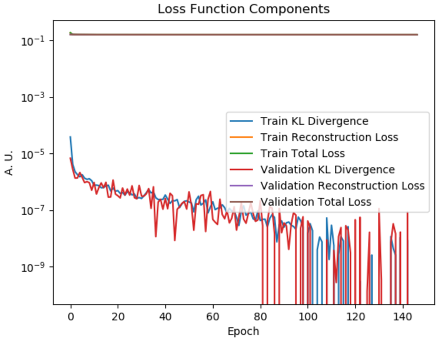
- 100p gluon jets

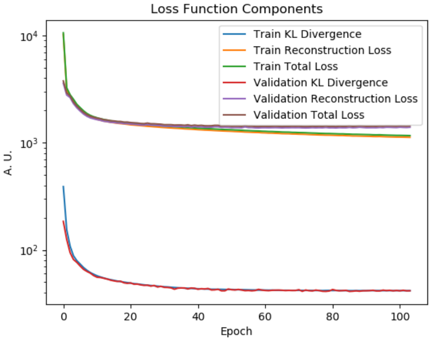


# Loss plots


Loss Function Components
A. U.
Epoch
Train KL Divergence
Train Reconstruction Loss
Train Total Loss
Validation KL Divergence
Validation Reconstruction Loss
Validation Total Loss


Loss Function Components
A. U.
Epoch
Train KL Divergence
Train Reconstruction Loss
Train Total Loss
Validation KL Divergence
Validation Reconstruction Loss
Validation Total Loss

# Backup

# Neural Network for the jets dataset

- In the Conv layers: kernel = (1,5) ((3,5) only in input and output layers), stride = (1), padding = (0,1); the order is $channels \times rows \times columns$ (also for ConvTranspose)
- No dropout or pool (for now)
- $ReLU$ activation function after all Conv and ConvTranspose (the output is an exception), the first dense layer, and all dense layers that appear after the latent vector